Irreducibla Markovkedjor

gean piere ventura cruz

2022-02-17

Uppgift 1

a. Visuell tolkning av sannolikhetsfördelning för dess tillstånd under flera dagars perioder då automaten utan några fel. Sannolikheterna är avrundade till 4 decimaler.

```
df <- data.frame(Tid = c("På onsdag", "Om en vecka",
"Om två veckor", "Om tre månader"))
df <- cbind(df, rbind(c(0.000, 0.000, 0.048, 0.272, 0.432, 0.248), c(0.0024960, 0.0272736, 0.
1242672, 0.3153496, 0.2614016, 0.2711120), c(0.002581384, 0.02582675, 0.1163860, 0.3107785,
0.2724897, 0.2719376), c(0.002590674, 0.02590674, 0.1165803, 0.3108808, 0.2720207, 0.272020
7)))
names(df)[-1] <- paste0("Tillstånd ", 0:5)
knitr::kable(df, digits = 4, caption = "Tabell 1.1 tillståndsverktorer efter olika antal daga
r med start från tillstånd 5")
```

Tabell 1.1 tillståndsverktorer efter olika antal dagar med start från tillstånd 5

Tid	Tillstånd 0	Tillstånd 1	Tillstånd 2	Tillstånd 3	Tillstånd 4	Tillstånd 5
På onsdag	0.0000	0.0000	0.0480	0.2720	0.4320	0.2480
Om en vecka	0.0025	0.0273	0.1243	0.3153	0.2614	0.2711
Om två veckor	0.0026	0.0258	0.1164	0.3108	0.2725	0.2719
Om tre månader	0.0026	0.0259	0.1166	0.3109	0.2720	0.2720

b. Den här gången befinner sig automaten i tillstånd 3.

```
df <- data.frame(Tid = c("På torsdag", "Om en vecka",
"Om två veckor", "Om tre månader"))
df <- cbind(df, rbind(c(0.0048, 0.0336, 0.1155, 0.2787, 0.2396, 0.3278), c(0.0023988, 0.02380
80, 0.1105299, 0.3060033, 0.2849188, 0.2723412), c(0.002600479, 0.02596195, 0.1166557, 0.3108
168, 0.2717323, 0.2722328), c(0.002590674, 0.02590674, 0.1165803, 0.3108808, 0.2720207, 0.272
0207)))
names(df)[-1] <- paste0("Tillstånd ", 0:5)
knitr::kable(df, digits = 4, caption = "Tabell 1.2 tillståndsverktorer efter olika antal daga
r med start från tillstånd 3")
```

Tabell 1.2 tillståndsverktorer efter olika antal dagar med start från tillstånd 3

Tid	Tillstånd 0	Tillstånd 1	Tillstånd 2	Tillstånd 3	Tillstånd 4	Tillstånd 5
På torsdag	0.0048	0.0336	0.1155	0.2787	0.2396	0.3278
Om en vecka	0.0024	0.0238	0.1105	0.3060	0.2849	0.2723
Om två veckor	0.0026	0.0260	0.1167	0.3108	0.2717	0.2722
Om tre månader	0.0026	0.0259	0.1166	0.3109	0.2720	0.2720

Tabellerna, 1.1 och 1.2 verkar konvergera mot ett gränsvärde för de olika tillstånd (0-5). Vi sedan märker att ju fler dagar vi itererar över kommer tabellernas rader att konvergera mot samma värde och oberoende av valt starttillstånd.

Uppgift 2

```
mpow <- function(P, n) {
  resultat <- diag(nrow(P))
  potens <- n
  while (potens > 0) {
    resultat <- P %*% resultat
    potens <- potens - 1
  }
  return(resultat)
}
P =  matrix(c(0, 0, 0, 0.5, 0, 0.5,
              0.1, 0.1, 0, 0.4, 0, 0.4,
              0, 0.2, 0.2, 0.3, 0, 0.3,
              0, 0, 0.3, 0.5, 0, 0.2,
              0, 0, 0, 0.4, 0.6, 0,
              0, 0, 0, 0, 0.4, 0.6),
            byrow = TRUE,
            nrow = 6,
            ncol = 6)
Q = matrix(c(0, 0, 0, 0.5, 0, 0.5,
              0.1, 0.1, 0, 0.4, 0, 0.4,
              0, 0.2, 0.2, 0.3, 0, 0.3,
              0, 0, 0.3, 0.5, 0, 0.2,
              0, 0, 0, 0.4, 0.6, 0,
              0, 0, 0, 0, 0.4, 0.6),
            byrow = TRUE,
            nrow = 6,
            ncol = 6)
```

a)Det verkar som att rader konvergerar mot samma värde då n = 20 då vi använder oss av funktionen rows_equal. När vi använder oss funktionen matrices_equal märker vi att P^(n)= P^(n+1) då n också är lika med 20.

```
rows_equal <- function(P, d = 4) {
  P_new <- trunc(P * 10^d) # förstora talet och ta heltalsdelen
  for (k in 2:nrow(P_new)) {
    if (!all(P_new[1, ] == P_new[k, ])) {
      return(FALSE)
    }
  }
  return(TRUE)
}
c=mpow(P,20)
rows_equal(c,d = 4)
```

```
## [1] TRUE
```

```
matrices_equal <- function(P, Q, d = 4) {
  P_new <- trunc(P * 10^d)
  Q_new <- trunc(Q * 10^d)
  if (all(P_new == Q_new)) {
    return(TRUE)
  } else {
    return(FALSE)
  }
}
p=mpow(P,20)
q=mpow(Q,21)
matrices_equal(P,q,d=4)
```

```
## [1] FALSE
```

Ett exempel på en sådan rad som konvergerar ser ut på följande sätt:

```
df <- data.frame(n = c("20"))
df <- cbind(df, rbind(c(0.002590259, 0.02590481, 0.1165789,0.3108859, 0.2720300, 0.2720100)))
names(df)[-1] <- paste0("Tillstånd ", 0:5)
knitr::kable(df, digits = 4, caption = "Tabell 2 tillståndsvektor efter 20 dagar")
```

Tabell 2 tillståndsvektor efter 20 dagar

n	Tillstånd 0	Tillstånd 1	Tillstånd 2	Tillstånd 3	Tillstånd 4	Tillstånd 5
20	0.0026	0.0259	0.1166	0.3109	0.272	0.272

b)Vi börjar med ange vår givna matris P som den sedan multipliceras med en vektor π_i för varje tillstånd. Som sedan adderas för att sedan få långtidsproportionen av övergångar från tillstånd i till j. Samt ska summan av alla långtidsproportionerna vara lika med 1. Vi kan då ställa upp följande funktion som löser denna typ av ekvation:

$$(P^t - I)\pi = 0$$

```
P =  matrix(c(0, 0, 0, 0.5, 0, 0.5,
              0.1, 0.1, 0, 0.4, 0, 0.4,
              0, 0.2, 0.2, 0.3, 0, 0.3,
              0, 0, 0.3, 0.5, 0, 0.2,
              0, 0, 0, 0.4, 0.6, 0,
              0, 0, 0, 0, 0.4, 0.6),
            byrow = TRUE,
            nrow = 6,
            ncol = 6)
A= t(P)-diag(6)
A[5,] = rep(1,6)
y = c(0,0,0,0,1,0)
pi=solve(A, y)
pi
```

```
## [1] 0.002590674 0.025906736 0.116580311 0.310880829 0.272020725 0.272020725
```

Lösningen av matrisekvationen ovan ger övergångsmatrisens P:s stationära fördelning.

Uppgift 3

```
gen_sim <- function(x, P) {
  u <- runif(1)
  y <- 0
  test <- P[x + 1, 1]
  while (u > test) {
    y <- y + 1
    test <- test + P[x + 1, y + 1]
  }
  y
}

P =  matrix(c(0, 0, 0, 0.5, 0, 0.5,
              0.1, 0.1, 0, 0.4, 0, 0.4,
              0, 0.2, 0.2, 0.3, 0, 0.3,
              0, 0, 0.3, 0.5, 0, 0.2,
              0, 0, 0, 0.4, 0.6, 0,
              0, 0, 0, 0, 0.4, 0.6),
            byrow = TRUE,
            nrow = 6,
            ncol = 6)
set.seed(1)

simulera_kedja <- function(x, n, P) {
  results <- numeric(n + 1) # vektor av nollor, till en början
  results[1] <- x # första elementet är initialtillståndet
  for (i in 2:n) {
    results[i] <- gen_sim(results[i - 1], P) # simulera övriga element
  }
  # ta bort initialtillståndet och returnera vektorn med simulerade tillstånd
  results[-1]
}
results2 <- simulera_kedja(5, 1000, P)

barplot(table(results2),
        xlab = "Tillstånd",
        ylab = "Antal",
        main = "1000 simuleringar av en Markovkedja")
```

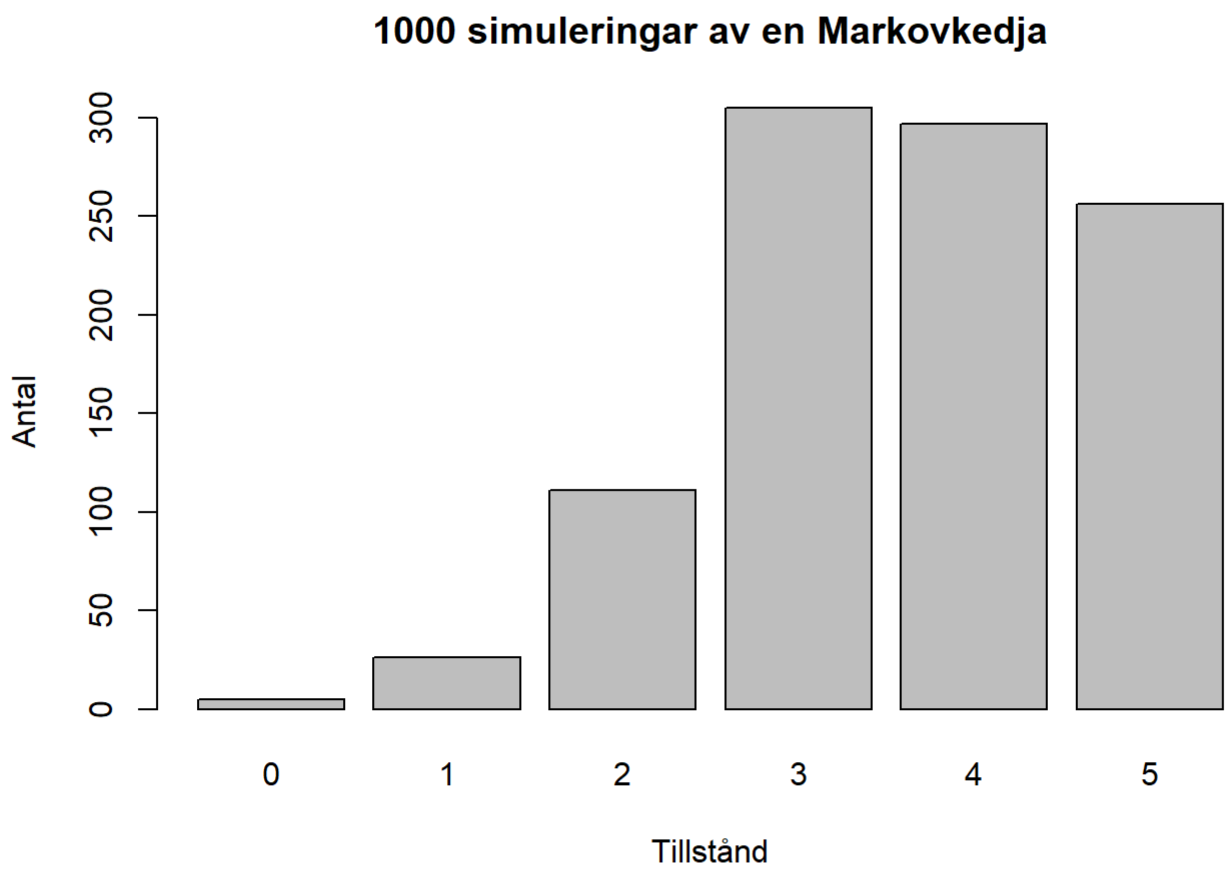

Diagram 1 visar en simulering av hur övergångsmatrisen fördelar dig i de olika tillstånd upp till dag 1000.

Vi märker genast att detta stämmer med den stationära fördelningen i tabell 2. Det går att se att i tabell 2 så är tillstånd 3 vanligast då den har högst sannolikhet att hända. Tillstånd 4 kommer på en andra plats och så vidare för resterande tillstånd. Detta resultat har då illustrerats i diagram 1.

Uppgift 4 a)

```
#functionen räknar ut hur många gånger vi är i tillstånd 5 och hur många gånger vi
#är i tillstånd 1 innan vi kommer till tillstånd 5.
tillstånd_5=0
tillstånd_1_5=0
for (i in 2:1000){
  if(results2[i] == 5){ #gånger vi är i tillstånd 5
    tillstånd_5 = tillstånd_5 + 1
  }
  if(results2[i]==5 && results2[i-1] == 1){ #gånger vi är i tillstånd 1 innan vi kommer till
tillstånd 5
    tillstånd_1_5 = tillstånd_1_5 + 1
  }
}
tillstånd_1_5/tillstånd_5
```

```
## [1] 0.03515625
```

b. Vi tillämpar Bayes formel, vilket ger oss följande formel:

$$P[(X_{n-1}=1)|(X_n)=5] = \frac{P[(X_{n-1}) \cap P(X_n=5)]}{P(X_n=5)} = \frac{P_{15} \cdot P(X_{n-1}=1)}{P(X_n=5)}$$

Sannolikhet att tillstånd 1 kommunicerar med 5 är 0.4, $P(X_n=5)=0.2720$ och $P(X_{n-1}=1)=0.0259$. Detta oss en sannolikhet på 0.038.